

Uso del operador swap genera soluciones eficientes computacionales en un caso de enrutamiento de vehículos con enfoque de ventanas de tiempo

The use of the swap operator generates efficient computational solutions in a vehicle routing case with a time window approach

DOI: <https://doi.org/10.17981/cesta.02.01.2021.05>

Artículo de investigación científica. Fecha de recepción: 05/03/2021. Fecha de aceptación: 23/06/2021.

Javier Darío Mantilla Mejía 

Universidad Santiago de Cali. Cali (Colombia)
javier.mantilla00@usc.edu.co

Para citar este artículo:

J. Mantilla, “Uso del operador swap genera soluciones eficientes computacionales en un caso de enrutamiento de vehículos con enfoque de ventanas de tiempo”, *J. Comput. Electron. Sci.: Theory Appl.*, vol. 2, no. 1, pp. 51–60, 2021. <https://doi.org/10.17981/cesta.02.01.2021.05>

Resumen

Introducción— La programación de ruteo de vehículos con cumplimiento de servicio es una necesidad de las empresas de logística en busca de su ventaja competitiva.

Objetivo— El objetivo del siguiente trabajo es determinar el ruteo de vehículos con ventanas de tiempo para una flota homogénea aplicado a un caso de distribución última milla con 300 clientes, considerando la minimización de los costos operativos, costos de distribución y costos de tiempos de inactividad.

Metodología— Se aborda el problema a través del planteamiento de un modelo matemático de programación lineal entera mixta, y el desarrollo de un algoritmo mediante uso del método de ahorros y el uso del operador swap.

Resultados— En la fase de construcción, el algoritmo de ahorros logra un costo inicial enfocado en la distancia mínima. En la fase de mejoramiento, el operador swap mejora la solución inicial establecida, de forma muy rápida. Para un caso de 300 clientes, se realizaron 12 iteraciones obteniendo una mejora del 71.41% sobre el costo inicial.

Conclusiones— Para cálculos de casos de VRPTW con 300 nodos, el operador swap consigue tiempos computacionales menores a 30 segundos.

Palabras clave— Aplicaciones VRP; Heurística; Matriz de Ahorro; Optimización Combinatoria; VRPTW

Abstract

Introduction— Vehicle routing scheduling with service compliance is a necessity for logistics companies in search of their competitive advantage.

Objective— The objective of the following work is to determine the routing of vehicles with time windows for a homogeneous fleet applied to the last, mile distribution case with 300 clients, considering the minimization of operating costs, distribution costs and, downtime costs.

Methodology— The problem is approached through the approach of a mixed-integer linear programming mathematical model, and the development of an algorithm through the use of the savings method and the use of the swap operator.

Results— In the construction phase, the savings algorithm achieves an initial cost focused on the minimum distance. In the upgrade phase, the swap operator improves the initially established solution, very quickly. For a case of 300 clients, 12 iterations were carried out, obtaining an improvement of 71.41% over the initial cost.

Conclusions— For calculations of VRPTW cases with 300 nodes, the swap operator achieves computational times of less than 30 seconds.

Keywords— VRP Applications; Heuristic; Saving Matrix; Combinatorial Optimization; VRPTW



I. INTRODUCCIÓN

Actualmente, con la tecnología, el comercio electrónico, los despachos a domicilio, han incrementado la logística de entrega de última milla [1], la cual genera grandes volúmenes de pedidos para las empresas, lo que hace que los programas de generación de ruteo para vehículos sean complejos [2]. El servicio al cliente depende en gran medida del cumplimiento del servicio, la programación y la calidad del producto en la entrega [3]. La estrategia de la cadena de suministro en las empresas se enfoca en reducir los costos logísticos y aumentar la satisfacción del servicio [4]. Al establecer un modelo de enrutamiento de vehículos con ventanas de tiempo VRPTW (Vehicle Routing Problem with Time Windows), se garantiza el servicio al cliente, con la entrega del producto en el tiempo acordado; se genera una programación de rutas garantizando unos costos mínimos de operación y distribución, y se da mejor administración de la flota empleada [2]. Estos modelos propuestos, podrán ser empleados por pymes con 100 entregas diarias, a través del uso del método exacto; y hasta 300 clientes a través del uso del algoritmo.

El Problema del Ruteo de Vehículo (VRP) fue iniciado por Dantzig y Ramser en 1959, a través del Problema del Agente Viajero (TSP), que consiste en determinar la ruta más corta para atender a los clientes, comenzando en un depósito y regresando a él [5]. El VRP, es parte de los problemas tipo NP-HARD, cuyas soluciones no pueden resolverse en un tiempo polinómico razonable y se deben resolver a través del uso de la optimización combinatoria [6], [7]. En el VRP se debe cumplir con la visita al cliente dado por un solo vehículo, el vehículo debe comenzar y terminar en el depósito y cada ruta no debe exceder su capacidad total de vehículos [6].

El Problema de Generación de Rutas para Vehículos con Ventanas de Tiempo (VRPTW) es un problema de optimización bicriterio. La distancia recorrida y el número de vehículos utilizados deben reducirse. Los métodos de solución del VRPTW, presentan dos fases. En la primera fase se presenta una heurística de construcción y en la segunda fase se realiza una heurística de mejora iterativa. Los métodos de solución para el VRPTW, se pueden clasificar en optimización exacta y algoritmos heurísticos [8]. Se utilizan también metaheurísticas para escapar de los óptimo locales [9].

El concepto de VRPTW se introdujo en 1987, aplicando heurísticas, métodos exactos de la solución VRP, pero integrando el concepto de ventanas de tiempo [10]. Investigadores canadienses en 1998 proponen un algoritmo genético híbrido, a través de fases de recombinación y mutación, que logran cálculos de solución rápidos para 100 clientes, utilizando distancias euclidianas relativas [11]. En Alemania, para el año 1999, propusieron una solución al VRPTW mediante el uso de búsqueda tabú, realizando instancias para 100 clientes, los resultados se compararon con los propuestos por Solomon, con soluciones de alta calidad [9]. La búsqueda tabú es una de las técnicas efectivas para resolver el VRPTW [12]. La Universidad de Singapur en 2001, realizan un híbrido entre la búsqueda tabú y el algoritmo genético, con resultados que superan cada una de las heurísticas individualmente [13]. En 2002, investigaciones polacas realizaron un método de solución VRPTW utilizando recocido simulado, los resultados que obtuvieron en el número de vehículos y las distancias de la ruta en las instancias de Solomon, fueron eficientes [8]. Estudios de la Universidad de Hagen en Alemania para 2002, realizan un VRPTW de 1000 nodos, la primera fase se realizó mediante un algoritmo genético y la segunda fase mediante una búsqueda tabú [14]. Científicos de Finlandia en 2003, propusieron el manejo de una metaheurística determinista a través de 4 fases y estableciendo modificaciones a una búsqueda de vecindad variable reactiva, con resultados confiables [15]. Instituciones de Portugal en 2004, realizaron un algoritmo de dos fases. Una fase constructiva dada por la búsqueda de vecindad, y una segunda fase dada por el intercambio entre rutas e inserción de clientes en las rutas, los resultados fueron efectivos tanto en tiempos de cómputo como en distancia [16]. En 2005, estudios en conjunto de Israel y Noruega presentan un algoritmo híbrido, utilizando en la primera fase, búsqueda local guiada, y en la segunda fase, evolución de estrategias metaheurísticas, demostrando ser muy competitivo, con buenas soluciones y tiempos de cálculo razonables [17]. Evaluaciones de la India, en 2017, proponen un método basado en optimización de colonia de hormigas y algoritmos genéticos (ACO-GA). El GA otorga diversidad a la población, mientras que ACO es muy eficaz en la optimización. Los resultados fueron efectivos con pocas iteraciones [6]. En 2019, mejoraron los resultados de la empresa indonesia en donde realizaron la investigación, Indonesia en 2019 mediante el uso de ACO implementado a un problema de VRPTW [18]. China y Japón, en 2020, desarrollaron un algoritmo evolutivo híbrido multiobjetivo, los resultados frente a las instancias Solomon, muestran que es eficaz en la convergencia [4]. El mismo año, chinos y alemanes en cooperación, realizaron un algoritmo de inteligencia de enjambre híbrido para la construcción de soluciones iniciales, utilizando el algoritmo de sistema colonia de hormigas ACS y el algoritmo de lluvia de ideas ABS; y posteriormente mejoradas, a través del uso de inter rutas e intra rutas [19]. Publicaciones iraníes proponen un algoritmo de enfoque multiobjetivo

basado en el modelo de evolución de aprendizaje LEM, el cual consigue soluciones eficientes de manera muy rápida con pocas iteraciones [20]. Los autores de algunas instituciones de Malasia proponen un algoritmo genético, que brinda diversificación de búsqueda, complementado con un algoritmo RR de arruinar y recrear, que intensifica la búsqueda de la solución [21]. En 2021, en la UMM (Malasia) desarrollaron un algoritmo inspirado en colonia de abejas artificial ABC modificado E-MABC, que adicional al MABC, ofrece una búsqueda intensificada entre dos rutas, generando resultados óptimos [23].

II. DEFINICION DEL PROBLEMA

El modelo se define en un grafo (N, A) , consiste en un conjunto de nodos $N\{1, \dots, n\}$, donde 1 es el depósito donde inicia la ruta y n es también el depósito donde debe terminar el enrutamiento. Los clientes son establecidos en un conjunto $CLIENTES \{2, \dots, n-1\} \in N$. Los nodos de salida son representados por $NODSALIDA \{1, \dots, n-1\}$ y los de llegada por $NODLLEGADA \{2, \dots, n\}$. El conjunto de arcos $A = \{(i, j): i, j \in N, i \neq j\}$. El conjunto de vehículos $NVEH\{1, 2, \dots, k\}$, donde k es el número de vehículos que se utilizarán para satisfacer al conjunto de clientes. Cada uno de los vehículos contiene una capacidad máxima de carga CAP y una capacidad máxima del volumen $CAPV$. Cada camino desde un nodo i al nodo j se calcula utilizando una distancia euclidiana $DIST_{ij}$. El servicio S_{ik} , es el tiempo de llegada de cada vehículo a cada cliente i , y debe darse por un intervalo de ventanas de tiempo, siendo A_i el tiempo mínimo de llegada y B_i el tiempo máximo de llegada. Debe darse en cada ruta que $(A_1 \leq A_i \leq B_i \leq B_n)$. Cada cliente debe contener algunos artículos a distribuir, con un peso $CARGA_i$ y un volumen VOL_i , donde la suma de la carga y volumen de los vehículos por ruta no puede exceder la capacidad de los vehículos. El tiempo del servicio en cada Cliente $i \in CLIENTES$ está representado por el parámetro $TSERV$. El costo del tiempo de inactividad T_{ik} se establece si el vehículo no está viajando o realizando un servicio. La variable binaria X_{ijk} , corresponde a un valor de 1, si el vehículo k viaja del vértice i al j sobre el arco (i, j) , y es igual a 0 de otra manera. El $COSTOVEH$ corresponde a los costos operativos del vehículo. $COSTODIST$ a costo de distribución y DTC a costo de tiempo de inactividad. El $VRPTW$ puede expresarse matemáticamente como:

$$\begin{aligned} Min \text{ COSTO } (\$) = & \sum_{j=2}^{n-1} \sum_{k=1}^v X_{ijk} * \text{COSTOVEH} \\ & + \sum_{i=1}^N \sum_{j=1}^n \sum_{k=1}^v DIST_{ij} * X_{ijk} \\ & * \text{COSTO DIST} + \sum_{i=1}^{n-1} \sum_{k=1}^v T_{ik} * DTC \end{aligned} \quad (1)$$

Sujeto a:

$$\sum_{j=2}^n \sum_{k=1}^v X_{1jk} = 1 \quad \forall i \in CLIENTES \quad (2)$$

$$\sum_{i=1}^{n-1} X_{1jk} = 1 \quad \forall k \in NVEH \quad (3)$$

$$\sum_{i=1}^{n-1} X_{ink} = 1 \quad \forall k \in NVEH \quad (4)$$

$$\sum_{i=1}^{n-1} X_{iuk} = \sum_{j=2}^n X_{ujk} \quad \forall u \in CLIENTES, \forall k \in NVEH \quad (5)$$

$$\sum_{i=2}^{n-1} \sum_{j=2}^n (X_{ijk} * CARGA_i) \leq CAP \quad \forall k \in NVEH \quad (6)$$

$$\sum_{i=2}^{n-1} \sum_{j=2}^n (X_{ijk} * VOL_i) \leq CAPV \quad \forall k \in NVEH \quad (7)$$

$$\begin{aligned} \text{Si } X_{ijk} = 1 \text{ entonces } S_{jk} - T_{ik} - S_{ik} - (TSERV + DIST_{ij}) &= 0 \\ \forall i \in CLIENTES, \forall j \in NODLLEGADA, \forall k \in NVEH \end{aligned} \quad (8)$$

$$\begin{aligned} \text{Si } X_{1jk} = 1 \text{ entonces } S_{jk} - T_{ik} - S_{ik} - (DIST_{ij}) &= 0 \\ \forall i \in CLIENTES, \forall j \in NODLLEGADA, \forall k \in NVEH \end{aligned} \quad (9)$$

$$X_{ijk} * (S_{ik} - A_i) \geq 0$$

$$\forall i \in NODSALIDA, \forall j \in NODLLEGADA, \forall k \in NVEH \quad (10)$$

$$X_{ijk} * (B_i - S_{ik}) \geq 0$$

$$\forall i \in NODSALIDA, \forall j \in NODLLEGADA, \forall k \in NVEH \quad (11)$$

$$X_{ijk} \in \{0,1\} \quad (12)$$

$$S_{ik} \geq 0 \quad (13)$$

$$T_{ik} \geq 0 \quad (14)$$

La función busca minimizar los costos operativos, costos de distribución y costos del tiempo de inactividad. La restricción (2) asegura que se atienda a todos los clientes. Las restricciones (3) y (4) establecen que todas las rutas comienzan y terminan en el depósito. Restricción (5) evitar subtours. Las restricciones (6) y (7) son para el cumplimiento de la capacidad de carga y el volumen de los vehículos. La restricción (8) y (9) generan un equilibrio en los tiempos de salida, tiempo de inactividad, tiempos de viaje, tiempos de servicio y tiempos de llegada. Las restricciones (10) y (11) cumplen con las ventanas de tiempo.

Los métodos de solución exacta solucionan VRPTW a menor escala, a medida que aumenta sus clientes, igual su tiempo lo hace de forma exponencial. Después de 50 nodos no funcionan bien [19]. Para la validación del modelo MILP, se ingresan datos con 102 nodos al modelo matemático, tomados de las instancias de Solomon C101 [10]. El resultado de la distancia es de 829 que corresponde a los mejores resultados encontrados en http://www.bernabe.dorronsoro.es/vrp/index.html?/Problem_Instances/CVRPTWInstances.html. El ruteo se puede observar en la Fig. 1.

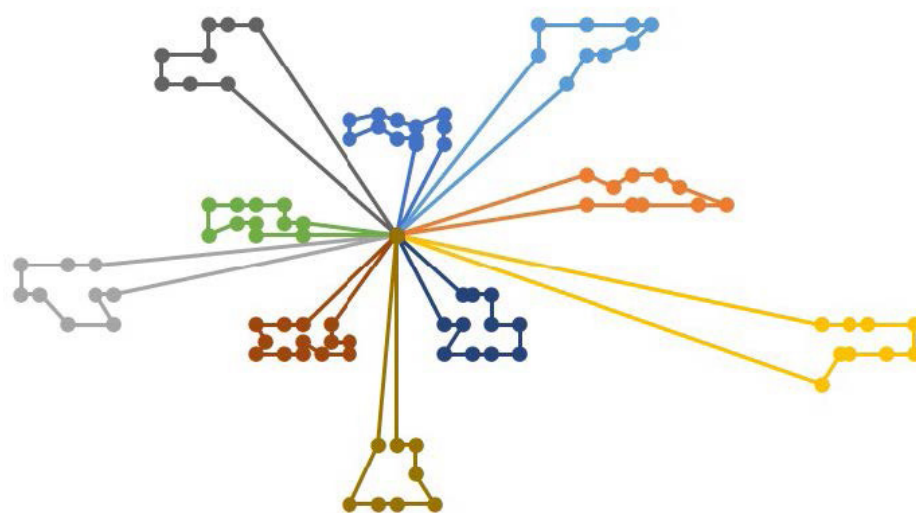


Fig. 1. Enrutamiento MILP caso 100 nodos instancia Solomon C101.
Fuente: Adaptado de [10].

III. DESCRIPCION DEL ALGORITMO

La primera fase del enrutamiento se realiza utilizando la matriz de ahorro [22] y posteriormente se mejora a través del uso del operador swap.

A. Algoritmo del Ahorro

Las coordenadas geométricas se utilizan para hacer la matriz de ahorro. La ecuación (15) fue utilizada por investigadores de Inglaterra en 1964 [22]. El máximo ahorro será el primero en ingresar al algoritmo de ahorro. Luego, es necesario verificar la restricción como peso, volumen y ventanas de tiempo. El costo debe calcularse en la primera ruta establecida.

$$Ahorro(i, j) = d(1, i) + d(1, j) - d(i, j) \quad (15)$$

El siguiente caso, se aplica al enrutamiento de 300 clientes. En la primera fase, la primera ruta se construye mediante el uso del algoritmo de ahorros. Se utilizaron 67 rutas. Se puede observar como las ventanas de tiempo mínimo y máximo son muy amplias (Fig. 4). En la segunda fase, el objetivo es reducir costos mediante el intercambio de rutas. Se puede ver cómo disminuyen las ventanas de tiempo (Fig. 5).

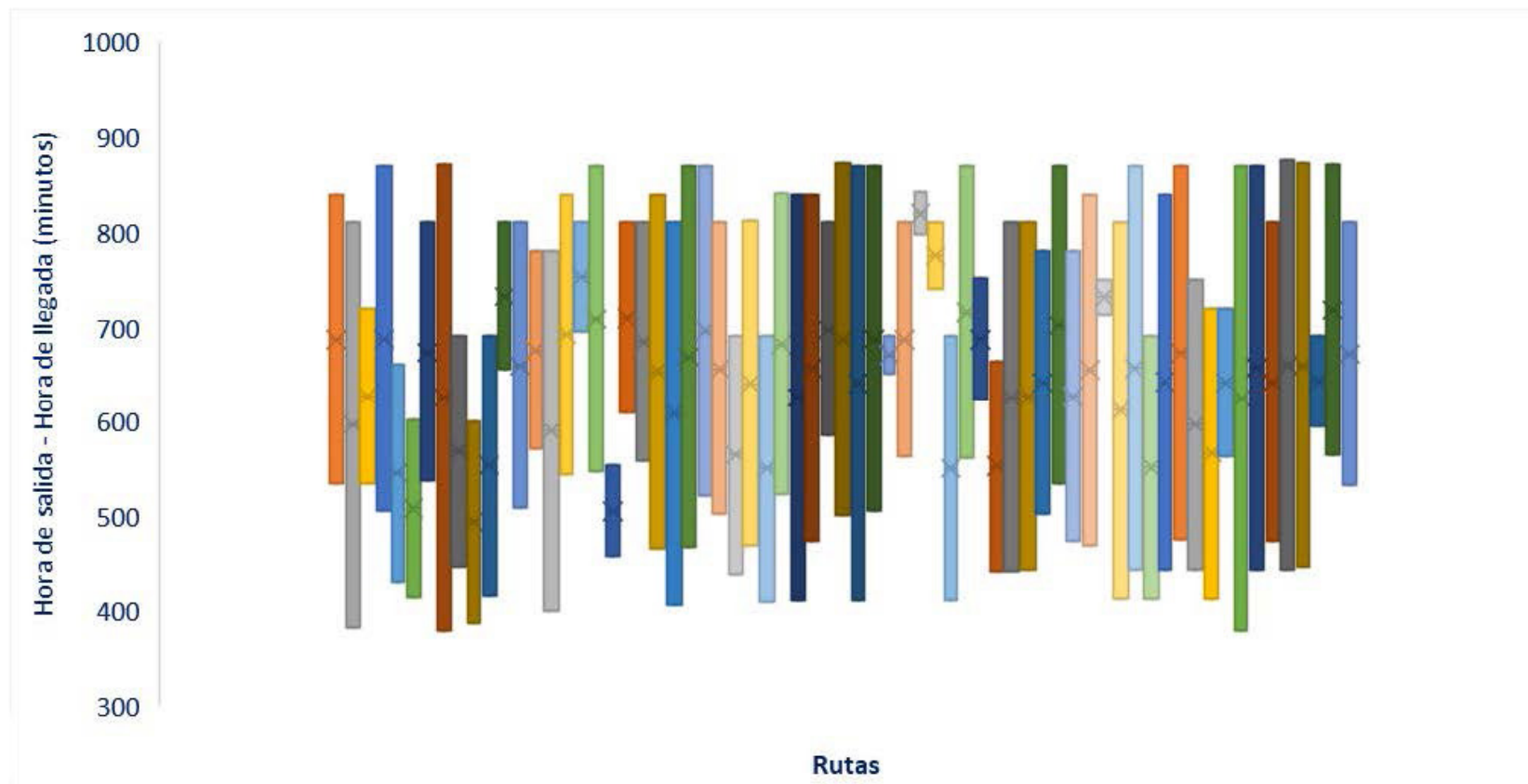


Fig. 4. Ventanas de tiempo 300 clientes - Algoritmo Ahorros.
Fuente: Autor.

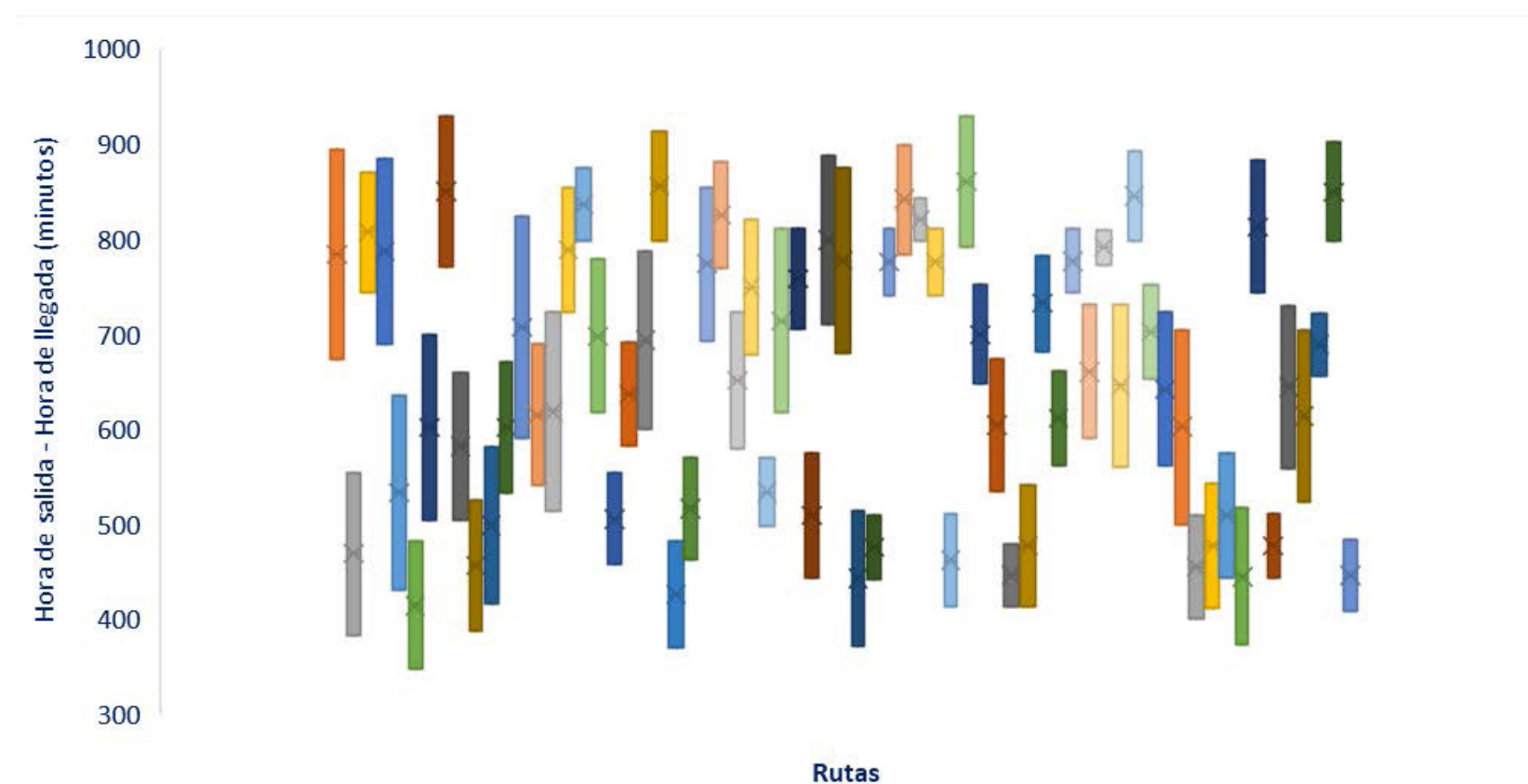


Fig. 5. Ventanas de tiempo - operador Swap.
Fuente: Autor.

IV. RESULTADOS

La matriz de ahorros se basa en la creación de una fase de construcción de rutas basada en la distancia, pero no tiene en cuenta el tiempo de inactividad de los viajes. El algoritmo de intercambio mejora el costo de la ruta, reduce el tiempo de inactividad mediante la mejora de los tiempos de salida, aprobando los mejores intercambios (Fig. 6).

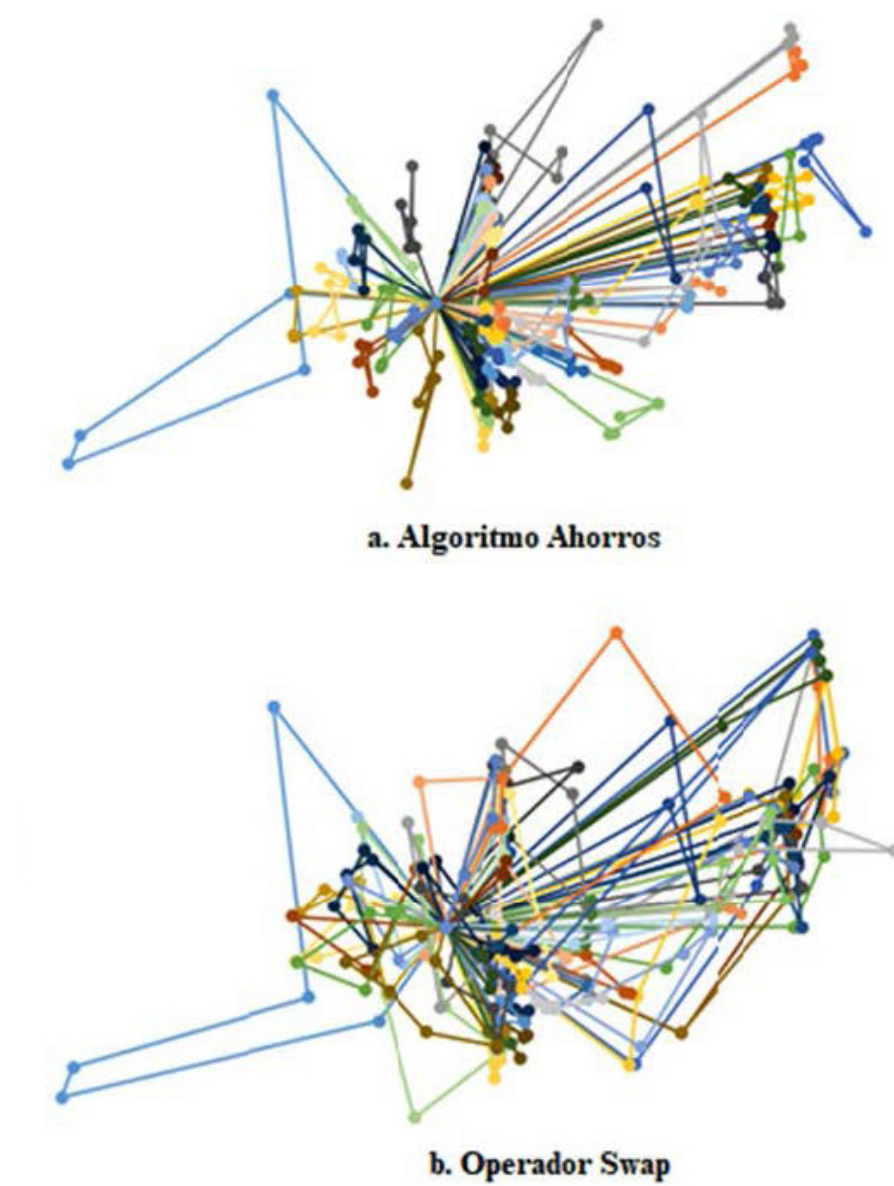


Fig. 6. a) Ruteos fase construcción algoritmo ahorros - b) Fase mejora Swap.
Fuente: Autor.

El algoritmo se programó en lenguaje C ++ y las pruebas computacionales se realizaron en un procesador Intel Celeron de 2,4 GHz. El resultado del tiempo del algoritmo es aceptable, para el caso de estudio en 100 Clientes tiene un dato de 0.629 segundos (Fig. 7). Esto muestra que el algoritmo de ahorros genera una búsqueda local [9], el algoritmo de intercambio genera una mejora rápida de los costos a través de un intercambio.

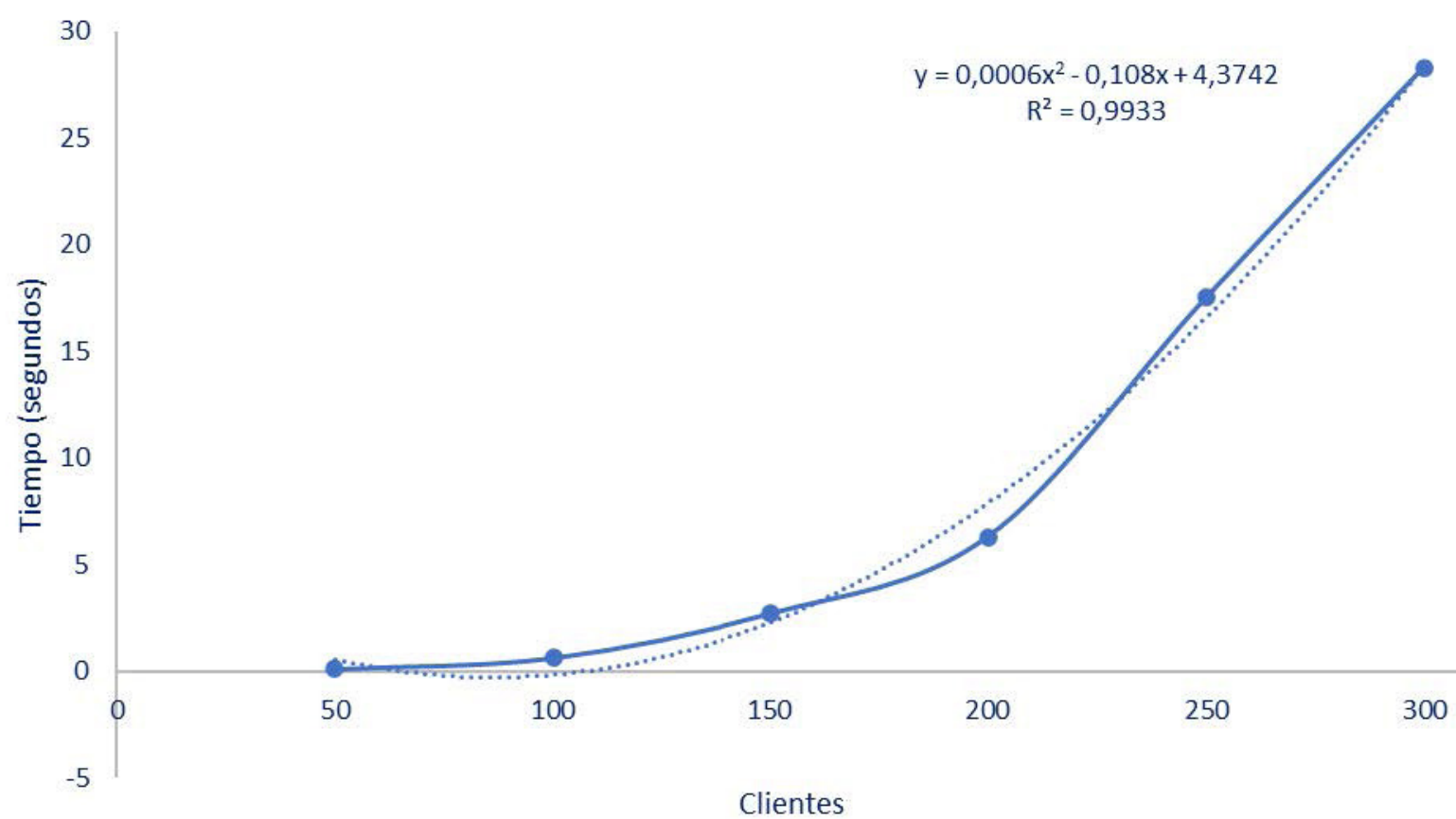


Fig. 7. Tiempos computacionales algoritmo Ahorro – Swap.
Fuente: Autor.

El tiempo computacional para establecer el enrutamiento de 300 clientes es de 28 644 segundos, de los cuales el algoritmo de ahorro toma 13 938 segundos, equivalente al 49.02% del tiempo total. El costo inicial del algoritmo de ahorro es de US\$ 517.68, y con el primer intercambio entre rutas, disminuyó en un 42.63%. En las siguientes 6 interacciones del algoritmo swap, se logra una mejora de costos del 71.08%, del total equivalente al 71.41% a un costo de US\$ 148.02 (Fig. 8).

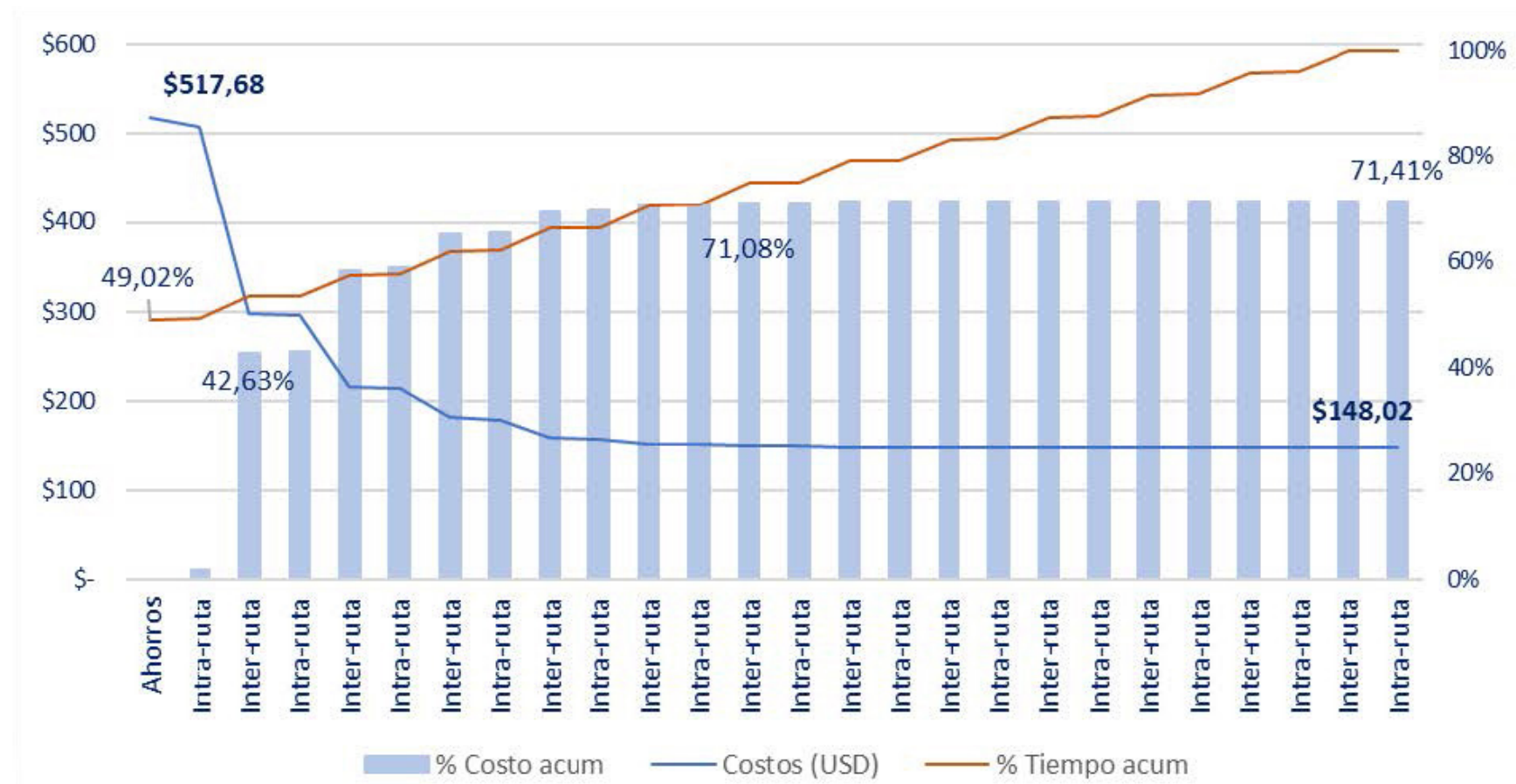


Fig. 8. Costo vs tiempo algoritmo Ahorros – Swap.

Fuente: Autor.

Se realizó un comparativo con las instancias C1, desarrolladas por autores alemanes en 2002 [14], las cuales vienen desarrolladas para 200 nodos incluyendo el depósito, con 50 vehículos con capacidades de 200 y un tiempo de servicio de 90. Cada nodo contiene sus coordenadas X, Y, demanda y ventanas de tiempo. Dentro los objetivos de estas instancias está en primer lugar disminuir el número de vehículos, y en segundo lugar disminuir la distancia recorrida. Se anexan resultados y las mejores soluciones dadas por la url <https://www.sintef.no/projectweb/top/vrptw/homberger-benchmark/200-customers/> (Tabla 1).

TABLE 1.

RESULTADOS COMPARATIVOS CON LAS INSTANCIAS C1 PARA 200 CLIENTES.

Instancia	Mejor solución		Ahorros - Swap			GAP
	#veh	distancia	#veh	distancia	tiempo(s)	
C1 201	20	2704.57	22	3007.01	0.767	11.2%
C1 202	18	2917.89	53	5181.68	1.259	77.6%
C1 203	18	2707.35	22	3007.01	0.753	11.1%
C1 204	18	2643.31	50	5050.78	3.740	91.1%
C1 205	20	2702.05	22	3036.44	0.966	12.4%
C1 206	20	2701.04	27	3623.84	1.208	34.2%
C1 207	20	2701.04	23	3133.21	0.999	16.0%
C1 208	19	2775.48	21	2832.02	1.499	2.0%
C1 209	18	2687.83	21	3006.69	1.069	11.9%
C1 210	18	2643.51	22	2978.37	1.367	12.7%

Fuente: Autor adaptada de instancias C1 de [14].

A diferencia con otros autores que han desarrollado modelos de solución con el uso del operador Swap, sus métodos incluyen operadores de inter rutas e intra rutas de forma aleatoria o con el uso de probabilidades [16], integran otros operadores en la fase de mejoramiento [21] [19].

V. CONCLUSIONES

El uso del operador swap, genera intercambios inter rutas y entre rutas de manera muy efectiva y con un número bajo de iteraciones. Se obtienen disminuciones en el costo de la fase de construcción hasta en un 71.41% para el caso presentado de 300 clientes, con un tiempo computacional. de 28 305 segundos.

El algoritmo Ahorros-Swap, a medida que aumenta el número de clientes, aumenta su tiempo de cómputo en un grado polinómico 2. El algoritmo de ahorro, para el caso de estudio presentado, ocupó casi la mitad del tiempo de cómputo para 300 clientes. El algoritmo Swap en 6 iteraciones de las 12 realizadas, obtuvo el 71.08% del costo inicial de la fase de construcción obtenido mediante el algoritmo de ahorros.

En las instancias presentadas C1 de Gehring & Homberger's, se puede concluir que en las instancias en donde el número de vehículos del algoritmo ahorros – swap se acerca al número de vehículos presentados en los mejores resultados, se presentan los menores GAP; y el método de ahorros es un referente inicial adecuado para comparar los datos con otras formas de obtener mejores resultados.

TRABAJOS FUTUROS

Futuros trabajos se orientarán a mejorar la fase de construcción a través de disminuir el número de rutas, a través del uso de heurísticas o metaheurísticas que se presentan en diferentes investigaciones. Ampliar el número de nodos del caso en estudio, y mejorar los GAP que se están presentando en este trabajo.

REFERENCIAS

- [1] W. Li, K. Li, P. N. R. Kumar & Q. Tian, "Simultaneous product and service delivery vehicle routing problem with time windows and order release dates," *Appl Math Model*, vol. 89, Part 1, pp. 669–687, jan. 2021. <https://doi.org/10.1016/j.apm.2020.07.045>
- [2] A. Moura & J. F. Oliveira, "An integrated approach to the Vehicle Routing and Container Loading Problems," *OR Spectr*, vol. 31, no. 4, pp. 775–800, 2009. <https://doi.org/10.1007/s00291-008-0129-4>
- [3] A. Expósito, J. Brito & J. A. Moreno, "Quality of service objectives for vehicle routing problem with time windows," *Appl Soft Comput JI*, no. 84, p. 105707, 2019. <https://doi.org/10.1016/j.asoc.2019.105707>
- [4] W. Zhang, D. Yang & G. Zhang, "Hybrid multiobjective evolutionary algorithm with fast sampling strategy-based global search and route sequence difference-based local search for VRPTW," *Expert Syst Appl*, vol. 145, pp. 113–151, 2020. <https://doi.org/10.1016/j.eswa.2019.113151>
- [5] G. B. Dantzig & J. H. Ramser, "The Truck Dispatching Problem," *Inform*, vol. 6, no. 1, pp. 80–91, 1959. <https://doi.org/10.1287/mnsc.6.1.80>
- [6] J. Gupta & C. Diwaker, "Evaluation of Capacitated Vehicle Routing Problem with Time Windows using ACO-GA," *IJRCSSE*, vol. 7, no. 6, pp. 610–615, 2017. <https://doi.org/10.23956/ijarcsse/V7I6/0319>
- [7] R. Baños & J. Ortega, "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows," *CAIE*, vol. 65, no. 2, pp. 286–296, 2013. <http://dx.doi.org/10.1016/j.cie.2013.01.007>
- [8] Z. J. Czech & P. Czarnas, "Parallel simulated annealing for the vehicle routing problem with time windows," presented at *10th Euromicro Workshop on Parallel Distributed and Network-based Processing, PDP, CI, Es*, pp. 376–383, 9-11 Jan. 2002. <http://dx.doi.org/10.1109/EMPDP.2002.994313>
- [9] J. Schulze & T. Fahle, "A parallel algorithm for the vehicle routing problem with time window constraints," *Ann Oper Res*, vol. 86, p. 585–607, 1999. <https://doi.org/10.1023/A:1018948011707>
- [10] M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window," *Oper Res*, vol. 35, no. 2, pp. 254–265, 1987. <https://doi.org/10.1287/opre.35.2.254>
- [11] J. Berger, M. Salois & R. Begin, "A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows," Conference presented at *Canadian AI 1998 Advances in Artificial Intelligence*, CSCSI, YVR, BC, Ca, 1998. https://doi.org/10.1007/3-540-64575-6_44
- [12] O. Braysy & M. Gendreau, "Tabu search heuristics for the vehicle routing with time windows," *BEIO*, vol. 10, no. 2, pp. 211–237, 2002. <https://doi.org/10.211-237.10.1007/BF02579017>
- [13] W.-K. Ho, J. Chin & A. Lim, "A hybrid search algorithm for the vehicle routing problem with time windows," *Int J Artif Intell Tools*, vol. 10, no. 3, pp. 431–449, 2001. <https://doi.org/10.1142/S021821300100060X>
- [14] H. Gehring & J. Homberger, "Parallelization of a Two-Phase Metaheuristic for Routing Problems with Time Windows," *J Heuristics*, vol. 8, pp. 251–276, 2002. <https://doi.org/10.1023/A:1015053600842>
- [15] O. Bräysy, "A Reactive Variable Neighborhood Search for the Vehicle Routing Problem with Time Windows," *JOC*, vol. 15, no. 4, pp. 347–368, 2003. <https://doi.org/10.1287/ijoc.15.4.347.24896>
- [16] A. Moura & J. F. Oliveira, "Uma heurística composta para a determinação de rotas para veículos em problemas com janelas temporais e entregas erecolhas," *Inv Op*, vol. 24, pp. 45–62, 2004. Extraído de apdio.pt/documents/10180/15407/IOvol24n1.pdf
- [17] D. Mestera & O. Braysy, "Active guided evolution strategies for large-scale vehicle routing problems with time windows," *Comp Oper Res*, vol. 32, no. 6 p. 1593–1614, 2005. <https://doi.org/10.1016/j.cor.2003.11.017>

- [18] I. Soenandi & B. Marpaung, “Capacitated Vehicle Routing Problem with Time Windows dengan Menggunakan Ant Colony Optimization,” *JSMI*, vol. 3, no. 1, pp. 59–66, 2019. <https://doi.org/10.30656/jsmi.v3i1.1469>
- [19] Y. Shen, M. Liu 1, J. Yang, Y. Shi & M. Middendorf, “A Hybrid Swarm Intelligence Algorithm for Vehicle Routing Problem With Time Windows,” *IEEE Access*, vol. 8, pp. 93882–93893, Mar. 2020. <https://doi.org/10.1109/ACCESS.2020.2984660>
- [20] B. Moradi, “The new optimization algorithm for the vehicle routing problem with time windows using multi-objective discrete learnable evolution,” *Soft Comput*, vol. 24, no. 9, p. 6741–6769, May. 2020. <https://doi.org/10.1007/s00500-019-04312-9>
- [21] T.-S. Khoo, B. B. Mohammad, V. Wong, Y.-H. Tay & M. Nair, “A Two-Phase Distributed Ruin-and-Recreate Genetic Algorithm for Solving the Vehicle Routing Problem With Time Windows,” *IEEE Access*, vol. 8, p. 169851–169871, Sep. 2020. <https://doi.org/10.1109/ACCESS.2020.3023741>
- [22] G. Clark y J. W. Wright, “Scheduling of vehicles from a central depot to a number of delivery points,” *Oper Res*, vol. 12, no. 4, pp. 568–581, 1964. <https://doi.org/10.1287/opre.12.4.568>
- [23] S. Mortada & Y. Yusof, “A Neighbourhood Search for Artificial Bee Colony in Vehicle Routing Problem with Time Windows,” *Int J Intell Eng Syst*, vol. 14, no. 3, pp. 255–266, 2021. <https://doi.org/10.22266/ijies2021.0630.22>

Javier Darío Mantilla Mejía es profesor adjunto del Departamento de Ingeniería Industrial en la Universidad Santiago de Cali (Colombia). Ingeniero Industrial. Especialista en Logística Integral. Magister en Ingeniería de la Universidad Javeriana (Cali, Colombia). <https://orcid.org/0000-0002-2018-7621>